

AMENDMENTS TO THE CLAIMS

Please amend the claims to be as follows, where markings are included to show changes made.

1. (currently amended) A ~~microprocessor~~ central processing unit (CPU) for targeted fault-tolerant computing, the ~~microprocessor~~ CPU comprising:
decode circuitry configured to decode a fault-tolerant version of an instruction and a non-fault-tolerant version of the instruction distinctly from each other; and
execution circuitry configured to execute the fault-tolerant version of the instruction with redundancy checking and to execute the non-fault-tolerant version of the instruction without redundancy checking.
2. (currently amended) The ~~microprocessor~~ CPU of claim 1, wherein the execution circuitry includes:
a first processing unit configured to receive operand data, execute an operation associated with the instruction; and generate a first result;
a second processing unit configured to receive the operand data; execute the operation, and generate a second result;
a comparator configured to compare the first and second results.
3. (currently amended) The ~~microprocessor~~ CPU of claim 2, wherein for the fault-tolerant version of the instruction, if the comparison does not match, then repeating the execution by the processing units and the comparison of results by the comparator up to a maximum N times until a match occurs.

4. (currently amended) The ~~microprocessor~~ CPU of claim 3, wherein, if the first and second results never match, a machine check is performed on the microprocessor.
5. (currently amended) The ~~microprocessor~~ CPU of claim 2, further comprising:
a register file configured to provide both the first and second processing units with the operand data.
6. (original) A method for targeted fault-tolerant computing in a central processing unit (CPU), the method comprising:
decoding a first op code corresponding to a fault-tolerant version of an instruction;
decoding a second op code corresponding to a non-fault-tolerant version of the instruction;
executing the first op code with redundancy checking; and
executing the second op code without redundancy checking.
7. (original) The method of claim 6, wherein a set of multiple instructions is provided in fault-tolerant and non-fault-tolerant versions of each instruction in the set.
8. (original) The method of claim 7, wherein the set of instructions includes arithmetic functions.
9. (original) The method of claim 7, wherein the set of instructions includes logical functions.
10. (original) The method of claim 6, wherein the execution of first op code comprises:
providing operand data to a first processing unit;

providing the operand data to a second processing unit;
executing an operation on the operand data by the first processing unit to
generate a first result;
executing the operation on the operand data by the second processing unit
to generate a second result; and
comparing the first and second results.

11. (original) The method of claim 10, further comprising, if the first and second results do not match, repeating the execution and comparison steps.
12. (original) The method of claim 11, wherein the repeating continues up to a maximum of N times until the first and second results match.
13. (original) The method of claim 12, further comprising, if the first and second results never matched during the N repetitions, performance of a machine check on the CPU.
14. (original) A computing apparatus for targeted fault-tolerant computing, the apparatus comprising:
means for decoding a first op code corresponding to a fault-tolerant version of an instruction and a second op code corresponding to a non-fault-tolerant version of the instruction;
redundant means for executing the first op code; and
non-redundant means for executing the second op code.
15. (original) The apparatus of claim 14, wherein the redundant means comprises:
a first processing unit configured to receive operand data, execute an operation associated with the first op code; and generate a first result;
a second processing unit configured to receive the operand data; execute the operation, and generate a second result;

a comparator configured to compare the first and second results.

16. (original) A computer program product comprising a computer-readable medium having computer-readable code embodied therein, the computer program product including:
 - a first type of computer-readable instructions to be executed with redundancy checking; and
 - a second type of computer-readable instructions to be executed non-redundantly.
17. (original) The computer program product of claim 16, wherein the first type of computer-readable instructions includes fault-tolerant arithmetic instructions.
18. (original) The computer program product of claim 17, wherein the second type of computer-readable instructions includes non-fault-tolerant arithmetic instructions.
19. (original) The computer program product of claim 16, wherein the first type of computer-readable instructions includes fault-tolerant logical functions
20. (original) The computer program product of claim 19, wherein the second type of computer-readable instructions includes non-fault-tolerant logical instructions.
21. (original) The method of claim 11, further comprising, if the first and second results do not match, logging a comparison error.